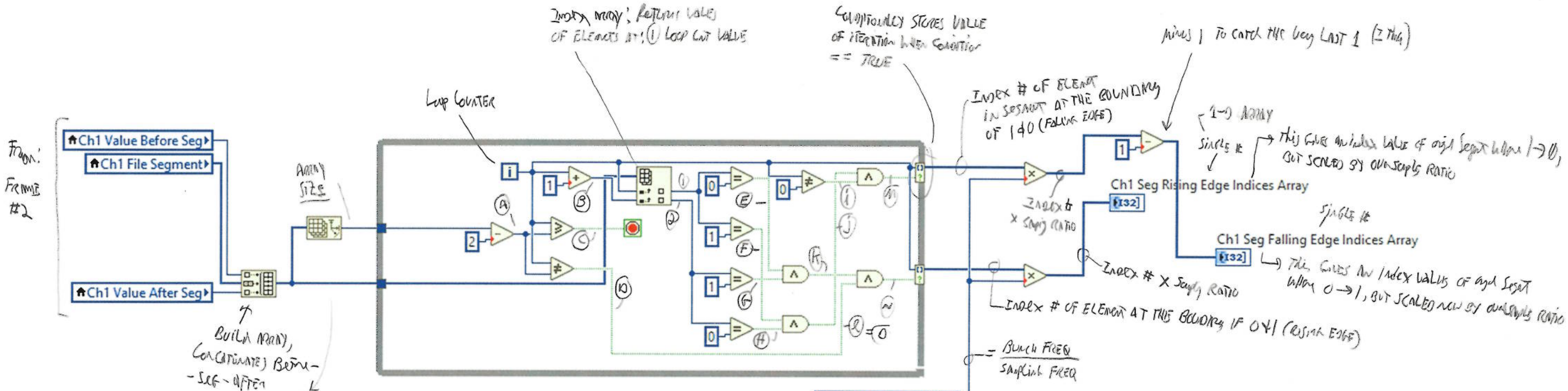
EVENT (5) - SEQ FRAME 3



VAL BEFORE	CH1 FILE SEGMENT	VAL AFTER

(A) Array Size − 2 = Ch1 File Segment Size

(B) Loop Cnt + 1

(C) If Loop Cnt ≥ Ch1 File Segment Size, then exit loop

(D) TRUE = Loop Cnt ≠ Ch1 File Seg Size

(E) TRUE IF: Array element at Loop Cnt N == 0

(F) TRUE IF: Array element at Loop Cnt N == 1

(G) TRUE IF: Array element at Loop Cnt N+1 == 1

(H) TRUE IF: Array element at Loop Cnt N+1 == 0

(I) TRUE IF: Loop Cnt ≠ 0

(J) TRUE IF: Array element at Loop Cnt N == 1
-AND-
Array element at Loop Cnt N+1 == 0
] Boundary Condition (1)

(K) TRUE IF: Array element at Loop Cnt N == 0
-AND-
Array element at Loop Cnt N+1 == 1
] Boundary Cond. (2)

(M) TRUE IF: Loop Cnt ≠ 0
-AND-
Boundary Cond (1) = TRUE

(N) TRUE IF: Loop Cnt ≠ Ch1 File Segment Size
-AND-
Boundary Cond (2) = TRUE

Q: (5) Input Segment Size == the Period of a Pulse (or Half-Period if seq only cycles)?

From EVENT 2: Req'd Iterations = $\frac{\text{Input Array Size}}{\text{Input Segment Size}}$ ← FP Control Setting

From EVT 1: # of Elements in Input Array →

⇒ This Segment of Code (EVENT[5] Seq Frame[2]) is used to extract a Segment of the Input Array (the Beam Pattern) → Local Var Array = CH1 File Segment → this is then used by downstream Events to build the WFM File that is loaded into the AWG. This code breaks the input pattern into chunks set by the GUI Input Segment Size Control

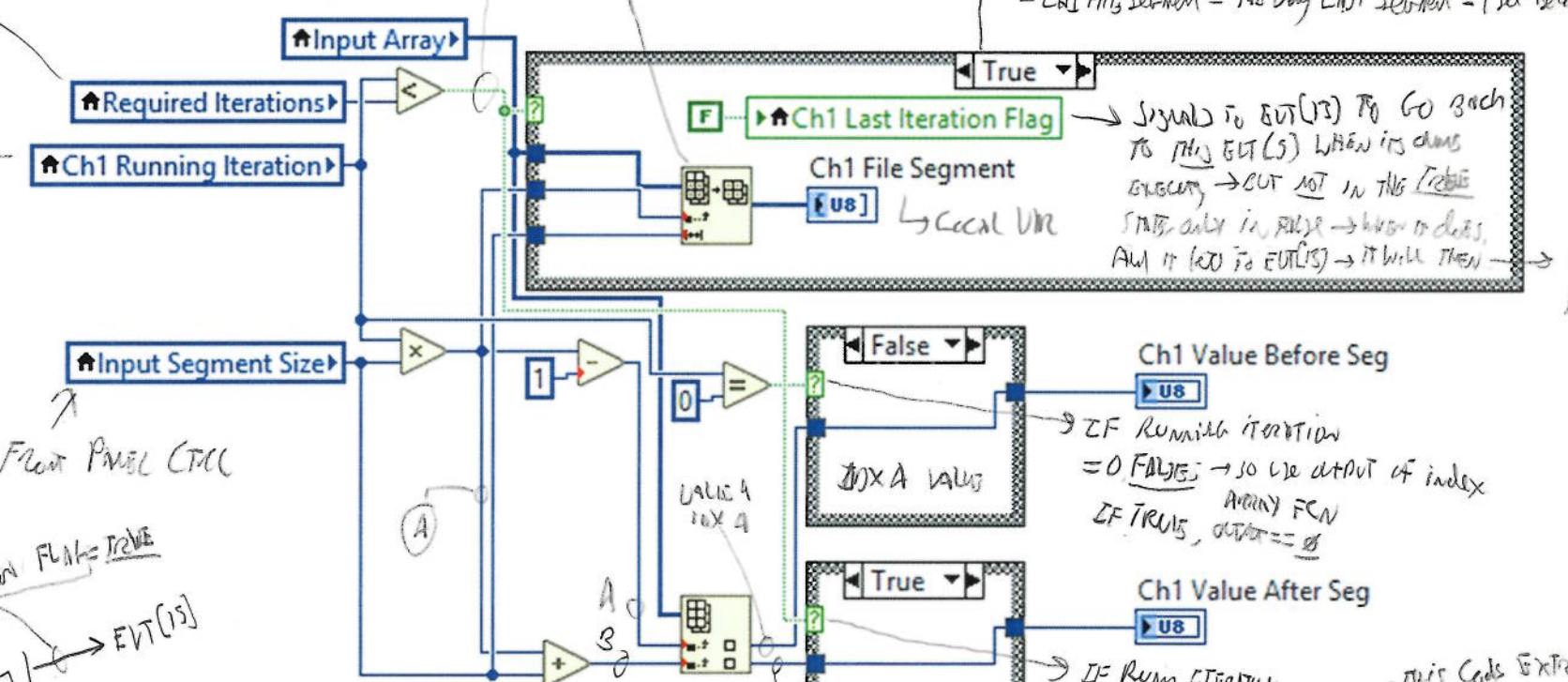
Array Pattern .CVF File

IF: CH1 Running Iteration < Required Iterations = True

Return a portion of the input array starting at index: Input Segment Size × CH1 Running Iteration

For length = Input Segment Size

→ When False: – CH1 Last Iteration Flag = True
– CH1 File Segment = the Very Last Segment = (See Below)



Signals to EVT[13] to go back to this EVT[5] when its done executing → but not in the Frame state only in False → when it does, and it goes to EVT[13] → it will then → goes EVT[15] & not back to EVT[5]

→ To next Frame(3) in Sequence

Init to 0 in EVENT[3] gets updated in EVENT[13] This is a Running Count This Count is in EVENT[13] It will keep counting until CH1 Last Iteration Flag = True → This is in Frame 3 of EVT[13] Sequence When Frame 3 executes, it forces EVENT[5] to execute again → but a bunch of other Events execute

Front Panel Ctrl

Value at idx A

Index Array Fcn

Return Element of Array at Input Index

Value at idx B

IF Running Iteration = 0 False → so use output of index Array Fcn
IF True, output == 0

IF Running Iteration = 0 = True → use output of Array Fcn, otherwise == 0

→ How do you "know" how long to make the input segment size??

Last Iteration Flag = True

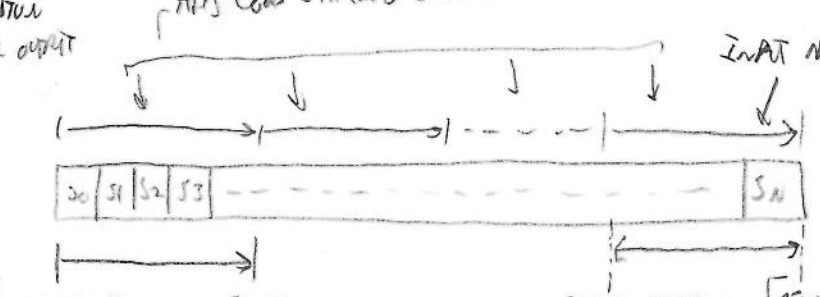
→ EVT[5] → EVT[7] → EVT[9] → EVT[11] → EVT[13] → EVT[15]

0 = False

IDX A = (CH1 Running Iteration × Input Segment Size) – 1
So gives value of array element → last element of previous segment

IDX B:
= (CH1 Running Iteration × Input Segment Size) + Input Segment Size
So gives the value of the array element just after the count segment → 1st element of next segment!

This Code extracts chunks of data from input array sequentially



(A) – CH1 Running Iteration × Input Segment Size

Input Segment Size

Required Iterations × Input Segment Size

Remaining Input Array Size (LVAR)

From EVT[2]:
= Remainder(0) $\frac{\text{Input Array Size}}{\text{Input Segment Size}}$

EVENT[5] → SEQ – Frame 2

→ What this section of code does is step thru the input array and extract a section (sub-section) of array data = Input Segment Size and stores it in the array CH1 File Segment

EVENT [15]

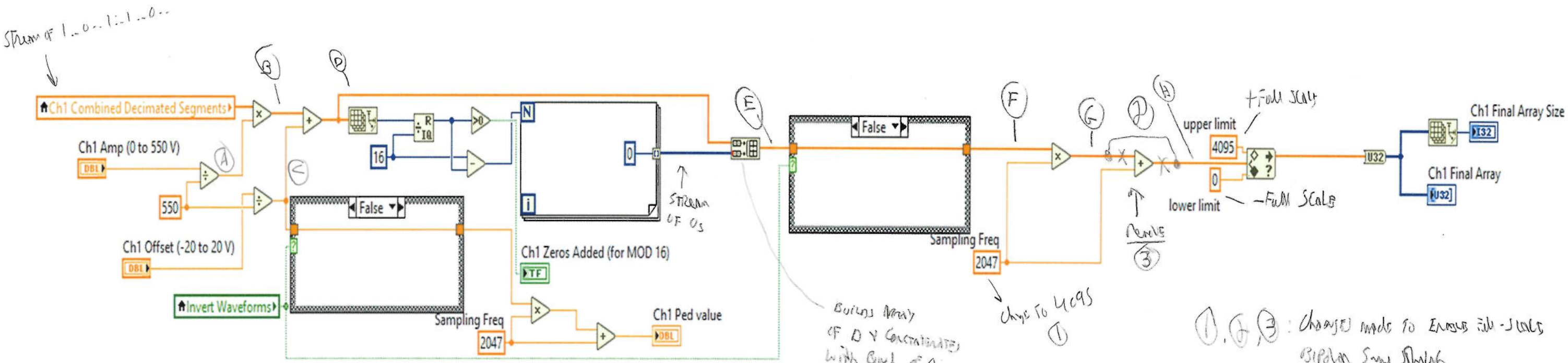
In Un-modified operation:

(H) = G + 2047

When G = 0 (input segment = 0), 1

Then (H) = 0 + 2047 = 2047 = 0V from DAC

What I did was change the constant to



1, 2, 3 : Changes made to enable full-scale bipolar signal sharing

D0 ... DN	0 ... 0m

(G) G = F × 2047

(A) If CH1 Amp = 550, A = 1

If CH1 Amp < 550, A < 1

A will range from 0 ... 1

(B) We multiply 1, 0 ... by A → so we get B = A or 0

(C) C = -20 ... 20/550

↳ Fractional value

(D) D = B + C

= (0 ... 1) + (Frac value) → D will only be bigger than 1.036

$\frac{20}{550} = 0.0\overline{36}$ or smaller